Universidade Federal do Rio de Janeiro

Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais

Sandro Roberto dos Santos Maciel

SOFTWARE-DEFINED NETWORK: O fim do monopólio comercial e incompatibilidades entre os fabricantes.

Rio de Janeiro

2016

Sandro Roberto dos Santos Maciel

SOFTWARE-DEFINED NETWORKING:

O fim do monopólio comercial e incompatibilidades entre os fabricantes.

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Orientador:

Cláudio Micelli de Farias, D.Sc., UFRJ, Brasil

Rio de Janeiro
2016

Sandro Roberto dos Santos Maciel

SOFTWARE-DEFINED NETWORKING:
O fim do monopólio comercial e incompatibilidades entre fabricantes.

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Aprovada em março de 2016.

Claudio Miceli de Farias, D.Sc., PPGI-UFRJ, Brasil

Dedico este trabalho aos meus filhos: Gabriela, Breno, Emanuelle e Vicente, por serem o motivo e a razão qual me obrigo diariamente a continuar na luta pelo progresso e resistente as dificuldades, sejam quais forem.

AGRADECIMENTOS

Gostaria de agradecer aos meus Orixás por injetarem doses de ânimos diários, a fim de me tornar um Guerreiro Vencedor, por me guiarem, me protegerem e abençoarem, mesmo em dias de turbulência. Todo esse privilégio em troca de Fé, Reconhecimento e muitas Rezas. OX!

RESUMO

MACIEL, Sandro Roberto dos Santos. **SOFTWARE-DEFINED NETWORKING: O fim do monopólio comercial e incompatibilidades em fabricantes**. Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2016.

Redes Definidas por Software tem como principal objetivo (além de melhorar o desempenho da Rede) a padronização na forma como os pacotes de dados são tratados nos dispositivos, independentemente de Fabricantes, os protocolos que controlam o fluxo, assim como o *Open Flow* e seus respectivos dispositivos que de forma inteligente se dividirem em funções especificas, promovendo melhor dedicação nas suas tarefas, são os grandes responsáveis por toda essa evolução, tratando e evitando diretamente os gargalos da Rede, que são provocados por Atraso de fila (buffer), processamento, tempo de transmissão e RTT *(Round-trip Delay Time).*

Apesar da forma como as Redes Definidas por Softwares com *Open Flow* tratam o fluxo de dados e controlam de forma relevante em comparação as Rede TCP/IP convencionais, existem pesquisadores que não as consideram uma revolução e sim uma evolução de outras técnicas que tem a centralização como principal objetivo, assim como: VoIP onde existe o Servidor Proxy e os usuários nas pontas, no *Wi-fi* tem a controladora com ponto central e os *AP's* remotos, e o *Firewall*, que possui suas regras de segurança também centralizadas e de onde veio a ideia de também centralizar o fluxo de dados, baseado no *Sane e Ethane*.

ABSTRACT

MACIEL, Sandro Roberto dos Santos. **SOFTWARE-DEFINED NETWORKING: O fim do monopólio comercial e incompatibilidades entre fabricantes**. Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2016.

Software Defined Network's main objective (besides improving network performance) is to standardize the way data packets are treated in network devices, regardless of Manufacturers, the protocols that control the flow, as well as the Open Flow and their respective devices that intelligently split into specific functions, promoting better dedication in its tasks, are the biggest responsible for all this evolution, treating and preventing network bottlenecks that are caused by queue delays (buffer), processing, transmission time and RTT (Round-trip Delay Time).

Despite the way Software Defined Networks with Open Flow treat the flow of data and a relevant form of control compared to the conventional TCP/IP, there are researchers who do not consider it a revolution, but an evolution of other techniques that have centralization as the main objective, just like VoIP where there is the proxy server and the client at the tips, Wi-fi has the centralized controller and the remote AP's, and the Firewall, which also has its centralized security rules and where the ideas of centralizing the data stream came, based on Sane and Ethane.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACL	Access-List
AP	Access Point
API	Appliation Programming Interface
ASIC	Application Specific Integrated Circuits
BPDU	Bridge Protocol Data Unit
BYOD	Bring Your Own Device
CLI	Command Line Interface
FIB	Forwarding Information Base
GRUB	Grand Unifield Bootloader
IEEE	Institute of Eletrical and Electronics Engineers
IOS	Internetwork Operating System
IoT	Internet of Things
IP	Internet Protocols
LDAP	Lightweigh Directory Access Protocol
LILO	Linux Loader
MPLS	Multi-Protocol Label Switching
NOS	Network Operating System
NOX	NOX
OSI	Open System Interconection
PCAP	Packet capture
QoS	Quality of Service
RIB	Routing Information Base
RTT	Round Trip Time
SDN	Software-Defined Networking
SSL	Secure Socket Layer
STP	Spanning-Tree Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TV Smarts	Television Smarts
VLAN	Virtual Local-Area-Network
VM	Virtual Machine
VOIP	Voice over Internet Protocol
WIFI	Wireless Fidelity

SUMÁRIO

1 INTRODUÇÃO

As Redes de comunicação estão crescendo em tamanho e complexidade em um ritmo cada vez maior, com a infraestrutura convencional, sistemas de redes e pilha de protocolos, dificilmente fornecem soluções adequadas para as demandas atuais de redes, além de possuir uma forma complexa e difícil de gerenciar vários tipos de equipamentos como: Roteadores, switches, firewalls, sistemas de detecção e prevenção de intrusos, balanceadores de carga e servidores, e todos na maioria das vezes pertencentes à fornecedores diversos, que disponibilizam ferramentas que auxiliam na configuração e monitoramento, onde não há compatibilidades entre as mesmas, com isso se faz necessário múltiplas plataformas de gerencia, dificultando na tratativa da falha, no levantamento da causa raiz do problema, na implantação de um projeto de ampliação da Rede, quando não disponibilizamos de uma única aplicação que suporta todo parque tecnológico, com certeza não aproveitamos o melhor de tudo que os fabricantes oferecem para os seus dispositivos. Estamos chegando no limite da escalabilidade e cada vez menos estáveis com as Redes atuais, começamos já algum tempo pelos endereçamentos de quatro Octetos, o tão conhecido IPv4, hoje no seu estágio final, chegando ao seu esgotamento, devido ao imenso crescimento de usuários na Internet, novas tecnologias já nem tanto futuristas, garantem que esses valores ainda sejam bem maiores, com a chegada da *IoT* (*Internet of Things*)[31], o propósito é que estejamos cada vez mais conectados, me refiro a usuários domésticos, pessoas que não fazem ideia de como e nem por onde conseguem acessar um site qualquer, nem mesmo aquelas que se quer detectam um cabo desconectado, agora, não apenas aos dispositivos (Laptop, Desktop, telefones), que estamos habituados à conectar em um AP (*Access Point)* ou até mesmo plugar nas tomadas de Redes, conhecidas como

RJ-45, mas os que fazem parte da nossa rotina, hoje já existem as *TVs Smarts*, rádios portáteis, carros e transportes públicos que se conectam via Wi-fi (o que não temos é Wi-fi disponível para esses veículos!), agora estamos falando da possibilidade de desligar a luz do quarto que esqueceu acessa de dentro da condução no caminho do trabalho ou até mesmo ligar o Ar-condicionado minutos antes de chegar em casa, olhar os itens na sua geladeira dentro mercado e não encarar fila para pagar suas compras no mesmo. Facilidades que a Internet tende a nos proporcionar, que elevará em altíssima escala o fluxo de dados, mais fabricantes e suas particularidades se integrando e impactando diretamente na operabilidade da Rede, maior numero de serviços distribuídos e cada vez mais difícil de operar tecnicamente. Isto provocou o surgimento de uma abordagem diferente para a arquitetura de sistemas de rede, *chamado Software-Defined Networking* (SDN).

1.1 OBJETIVOS.

Esta Seção apresenta os objetivos do presente trabalho. Assim, a Seção 1.1.1, introduz o Objetivo Geral dessa monografia e a Seção 1.1.2, descreve os Objetivos Específicos a serem alcançados.

1.1.1 Objetivo Geral.

Este trabalho tem como principal objetivo citar as dificuldades que operadores de Redes de Computadores encontram, quando atuam em um parque tecnológico, onde, há diversos produtos de fabricantes distintos interconectados (mesmo tendo um único objetivo), comparar as Redes Convencionais e as Redes Definidas Por

Software (SDN - *Software-Defined Network)* e suas formas de operar, mostrar as grandes tendências da Internet que a cada dia surgem para facilitar o nosso cotidiano e analisar o porquê de tanta escalabilidade e resiliência.

1.1.2 Objetivos Específicos.

- Apresentar sucintamente a Origem da SDN.
- Comparar Redes Convencionais e os Benefícios das Redes Definidas por Software.
- Apresentar a forma como as Redes Definidas por Softwares são implantadas.
- Apresentar o Open Flow como protocolo de comunicação e suas características.
- Apresentar algumas incompatibilidades entre Fabricantes.

2 CONCEITOS BÁSICOS

As Redes convencionais foram criadas na década de 1970 (elaborada inicialmente há 20 anos antes como forma e evitar a falta de comunicação na Guerra Fria, por conta da destruição em massa das Centrais Interurbanas que suportavam as Centrais de comutação telefônica [1].

Não se imaginava à proporção que a *Internet* tomaria ao longo de desses 46 anos de existência, naquela época 32 bits era consideravelmente suficiente para lidar com todo o Protocolo de Internet (IP) como endereçamento de Redes de Computadores, proporcionando o equivalente à 16 milhões (aproximadamente) de endereços usados para comunicação entre equipamentos, todo esse endereçamento foi nomeado como IPv4 (*Internet Protocol version 4)*, que recentemente em 2011, anunciaram o seu esgotamento. Além da expectativa do endereçamento IPv4, também não faziam menção à mobilidade e flexibilidade na estrutura de Rede, que uma vez estabelecida, a topologia não mudaria tanto.

Quando compramos Switches e Roteadores convencionais, praticamente compramos produtos prontos para o uso, programados e estruturados para determinados serviços específicos, havendo a necessidade de uma pesquisa aprofundada na decisão da compra do dispositivo, baseado na demanda oferecida e na estrutura que pretende ser atendida. Esses dispositivos são como caixas pretas fechados e lacrados não nos permitindo melhorar e/ou expandir seus componentes, é o que conhecemos como *OnBoard*, assim, somos forçados a gastos periódicos em dispositivos com melhor desempenho e escalabilidade, por conta de tecnologias futuras, que a cada dia vem tomando espaço no mercado como: *BYOD, IoT, Cloud Computing,* etc...., as taxas de fluxos de dados são cada vez maiores, exigindo dispositivos mais potentes com alta nível de processamento.

Os servidores que antes associavam cada aplicação à um único Sistema físico e exclusivo aos seus serviços, passaram por uma enorme transformação na última década. Com a chegada da virtualização, as formas de instalar e manusear fisicamente, os servidores mudaram drasticamente, ganharam dinamismo e mobilidade, não havendo a necessidade de reestruturação física em Rack's, instalações elétricas e cabeamento, hoje podemos usar o mesmo Sistema (físico) para várias aplicações diferentes, distribuindo em múltiplas Máquinas Virtuais, conhecidas *com VM (Virtual Machine)*, disponibilizando funções multitarefas sem que um processo interfira no outro, podendo as VM's se comunicarem ou não, ficando a cargo do Administrador.

2.1 PRINCIPIOS E TENDÊNCIAS.

Junto com a virtualização de servidores, muitas empresas também estão usando uma única rede para entregar serviços de voz, vídeo e dados, que em redes legadas de hoje, o conceito de Qualidade de Serviço (QoS) é usado para fornecer um nível diferenciado de serviço para diferentes aplicações. No entanto, o provisionamento de muitas ferramentas de QoS é altamente manual. Os Administradores da rede devem configurar o equipamento de cada fornecedor separadamente e ajustar os parâmetros, tais como a largura de banda e QoS por sessão, por aplicação, as Redes Convencionais são estáticas, e não conseguem se adaptar dinamicamente às novas exigências de tráfego, aplicações e usuários

As Rede Definida por Software surgiu em 2010/2011 na Universidade de Standford, criado por um aluno de Doutorado (Martin Casado) orientado por professores da Universidade de Berkeley (Scott Shenker) e Universidade de

Standford (Nick McKeown), e criaram uma *startup* chamada *Nicira*, quem criou o NOX (um sistema operacional de rede) e ao mesmo tempo, em conjunto com equipes da Universidade de Stanford criaram o switch com Interface OpenFlow, atualmente comprada pela *Vmware* por $1,26 Bilhões, desde 23 julho 2012, foi a pioneira em SDN [2].

Na Arquitetura SDN, tudo o que é inteligência de sistema operativo fica concentrado, onde haja "visibilidade global" sobre toda a rede. Portanto, ao invés de replicar todos os protocolos de roteamento em todos os dispositivos, eles ficam num só lugar. Com este sistema operacional implantado em plataformas abertas de servidores, linguagens e outros sistemas operacionais, a introdução de aplicações e funcionalidades torna-se uma questão de instalação de pequenos programas, elaborados para quem quiser programar, podendo ser um fabricante ou um operador da rede.

Várias aplicações de rede, tais como protocolos de roteamento utilizam sistema operacional e hardware de encaminhamento de pacotes para atingir os objetivos de roteamento, que para fazer alterações no comportamento da rede, é necessário acessar cada roteador na rede e emitir um conjunto de comandos, como finalidade principal mudar as características do roteador, em um sistema operacional e numa linguagem (fechada) que tenha sido definida por cada fornecedor, que por incompatibilidades podem não interagir facilmente com outros componentes que compõem a rede. Essas mesmas incompatibilidades estimularam o crescimento da SDN, pesquisas foram motivadas por problemas de comportamentos nas Redes atuais.

2.1.1 **Arquitetura das Redes Definidas por Softwares**

As Redes de Computadores hoje usam três planos separados para realizar tarefas: o plano de dados (ou plano de encaminhamento), o plano de controle e Gerenciamento.

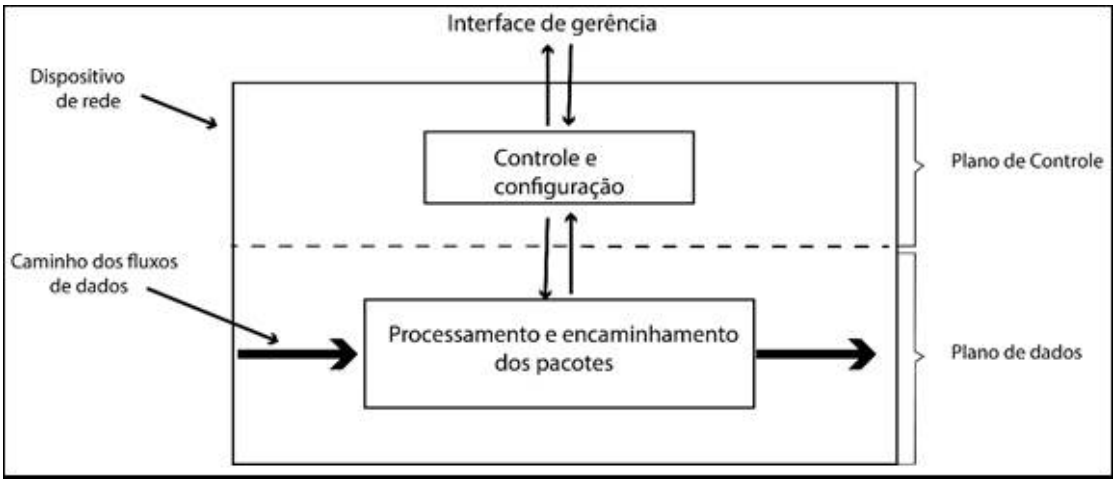

Figura 01 - Planos de Dados, Controle e Gerenciamento.

Redes Definidas por Software foram elaboradas com o propósito de separar o plano de dados do plano de controle, representando um grande avanço para o futuro das redes de dados. A separação entre os planos de controle e dados permite que a rede possa ser mais eficiente e barata, além de suprimir o problema encontrado nas redes legadas em que um determinado produto deve ser comprado de um mesmo fornecedor por meio de um pacote, contento no mesmo, o hardware e o software. Desta forma, a rede será formada por dispositivos “burros” encaminhadores de pacotes e por uma camada de software programável onde haverá um controlador de rede que coordenará todas as ações dos dispositivos. Esses dispositivos encaminhadores de pacotes serão programados através de uma interface aberta, como, por exemplo, a interface definida pelo *OpenFlow* [3].

2.1.2 **O Plano de Dados ou Encaminhamento (*FIB -Forwarding Information Base - (hardware - asic)).***

O Plano de Dados é o responsável pelo processamento dos pacotes recebidos. Quando um pacote chega, o Plano de Dados usa as informações sobre o estado de encaminhamento local e as informações que estão contidas no cabeçalho de cada pacote para tomar uma decisão como: descartar ou encaminhar o pacote. Se o plano de dados decidir enviar, então ele deve se certificar por qual switch e para qual porta no switch deve receber o pacote. Para manter o ritmo com todos os pacotes que estão chegando, o processamento do Plano de Dados deve ser feito de forma extremamente rápida.

Para otimizar o desempenho de encaminhamento dos roteadores existe outra estrutura de dados no plano de encaminhamento que é denominada FIB (*Forwarding Information Base*), sendo essa a mais próxima da tradicional tabela de roteamento que consultamos frequentemente no cotidiano. A FIB é uma estrutura bem mais simples que armazena somente a melhor rota para cada destino, ou algumas melhores, dependendo da inteligência de roteamento. Normalmente essa estrutura é implementada através de tabelas *hash* para torná-la menor e agilizar o processo de busca de informação (*lookup*).

2.1.3 **O Plano de Controle (RIB - *Routing Information Base* - (cérebro - IOS)):**

É no plano de controle que fica armazenada uma estrutura de dados denominada RIB (*Routing Information Base*). Essa estrutura é responsável por armazenar todas as informações de vizinhança entre os roteadores e todas as rotas conhecidas até um determinado destino, seja ela é a melhor ou não, simplesmente todas as rotas possíveis estão armazenadas nessa estrutura. É com base nessas

informações que agem os principais protocolos de roteamento, seja para adicionar, modificar, remover ou consultar seu conteúdo. É de se imaginar que essa estrutura normalmente não é pequena e que esse processo de implementar a inteligência da rede não é simples porque envolve a ação de algoritmos que podem ser bem complexos. A FIB (plano de encaminhamento) é gerada a partir da RIB (plano de controle). Dessa forma a operação do plano de encaminhamento ganha um certo grau de independência dos processos em execução no plano de controle e os dispositivos de rede podem fazer um *lookup* mais rápido e eficiente, o que reflete no desempenho do encaminhamento de pacotes. Em síntese, o plano de encaminhamento se limita a receber os pacotes, buscar uma correspondência na FIB e faz o encaminhamento dos pacotes se houver essa correspondência [4].

O plano de controle calcula o estado de encaminhamento que o Plano de Dados usa para encaminhar os pacotes. Este estado de encaminhamento pode ser calculado usando algoritmos distribuído ou algoritmos centralizados, ou pode ser configurado manualmente.

Com isso, o mercado passa a ser mais aberto, pois um determinado software de uma empresa fornecedora pode interagir com o hardware de outra, sem problema algum. Isso só é possível porque as Redes Definidas por Software usam um protocolo aberto como já citado, o protocolo OpenFlow por exemplo, que permite que a troca de informações entre o hardware e o software seja realizada sem que ambos estejam alocados no mesmo equipamento, de forma totalmente segura.

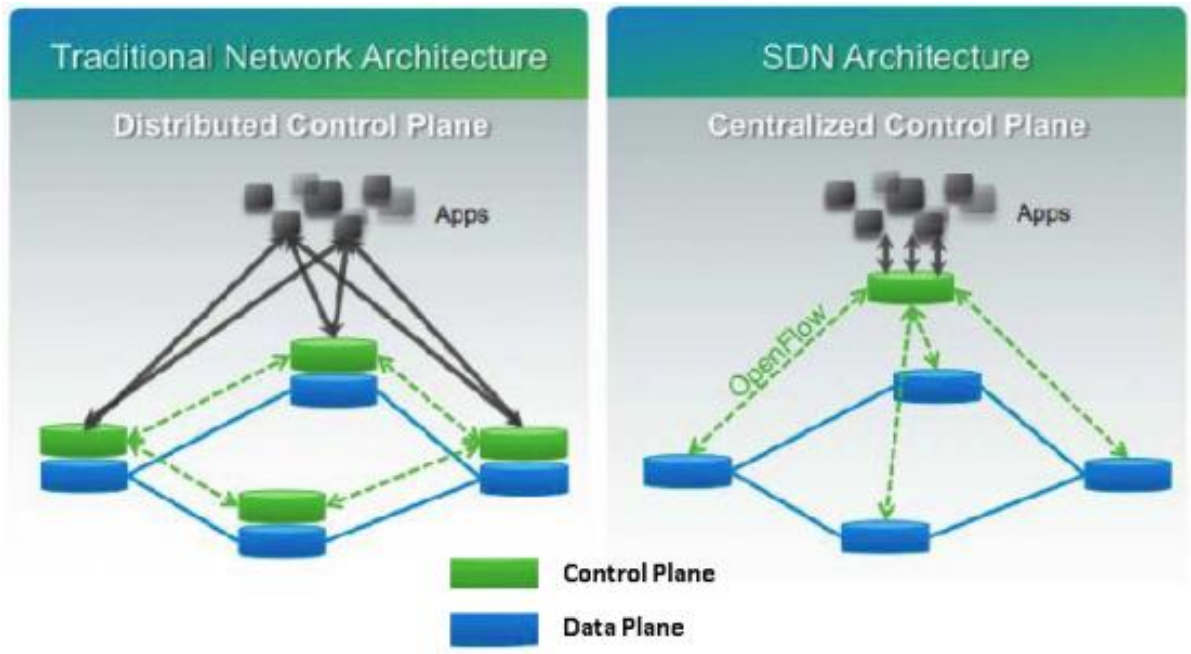

Figura 02 - Comparação entre o Plano de Dados e o Plano de Controle, das Redes Atuais e as SDN's.

Os mecanismos do plano de controle foram todos concebidos para realizar uma variedade de objetivos. Um exemplo são os planos de controle de roteamento que aplicam uma grande variedade de algoritmos de encaminhamento distribuído. Tem também planos de controle de isolamento que podem ser usados para fornecer listas de controle de acesso (*ACLs – Access Lists*), redes locais virtuais (*VLAN*), firewalls, e assim por diante. Existem planos de controle baseados em trafico de engenharia que interferem diretamente decisões de roteamento, similar com MPLS (*Multiprotocol Label Switchi*ng) [22]. Todas estas diferentes abordagens são tentativas de afetar o encaminhamento de pacotes, controlando o modo como o estado de encaminhamento é calculado.

O problema com os planos de controle usados nas redes atuais é que eles não são modulares, eles não podem ser utilizados em conjunto. Cada um dos planos de

controle resolve parte do problema, mas nenhuma deles resolvem todos, cada um fornece funcionalidade limitada.

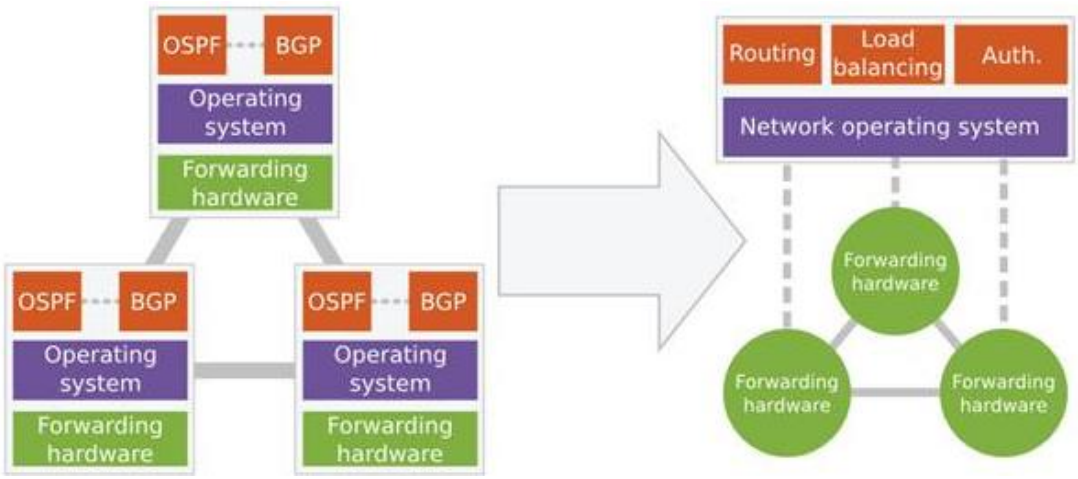

Figura 03 - Plano de Controle das Redes Atuais e em Redes Definidas por Softwares

2.1.4 **O Plano de Gerenciamento.**

É o responsável para coordenar a interação entre o plano de controle e plano de dados. Como todos os computadores, os dispositivos de rede precisam ser configurados e gerenciados. O plano de Gerenciamento fornece instruções básicas sobre como o dispositivo de rede deve interagir com o restante da rede. O plano de Controle pode aprender sozinho tudo o que precisa saber sobre a rede e o plano de Gerenciamento precisa saber o que fazer. Atualmente, os dispositivos de rede são configurados individualmente e com frequência, manualmente com o uso de uma interface de linha de comando (CLI – *Command Line Interface*) complexa e compreendida por um pequeno número de especialistas de rede. Como a configuração é manual, os enganos são frequentes e, às vezes, esses enganos têm sérias consequências como um corte de tráfego de todo um Data Center ou

interrupção de tráfego em uma importante rota do país. Os *Services Providers* se preocupam com as retroescavadeiras que cortam fisicamente seus cabos de fibra ótica, mas esquecem que há o corte virtual de cabos realizado por seus Operadores quando cometem um simples engano na complexa CLI usada para configurar os roteadores da rede ou *firewalls* de segurança.

3 **IMPLANTANTANDO SDN**

Embora grande parte da discussão sobre SDN envolve a switches de interfaces *OpenFlow* [23] na parte física das Redes Definidas por Softwares, o *OpenFlow* deve ser pensado a partir de um ponto de vista técnico como o componente menos interessante de SDN.

Ao contrário de protocolos de roteamento das redes distribuídas de hoje, SDN pode ser pensada como simplesmente computar funções e/ou serviços. Isto permite a SDN ignorar a infraestrutura física que tem sido usado para implementação de redes convencionais e permite que os Administradores gerenciem o tráfego de rede do plano de controle sem ser limitado pela estrutura física. Em SDN, o sistema operacional de rede (NOS - ***Network Operating System***) [24] é responsável por tomar decisões em funções especificadas e se certifica que os resultados das funções sejam distribuídos para cada comutador na rede.

Ao contrário da maioria dos aplicativos de nuvem, as redes convencionais são inerentemente descentralizadas. Isso é realmente tudo que as redes virtuais necessitam para mover dados de um ponto para outro. Assim, enquanto o *Facebook* pode ser acessível em um número pequeno de grandes datacenters, as redes são distribuídas ao longo de um datacenter, um campus, dentro de uma cidade, ou no caso da Internet, por todo o planeta. É por isso que redes sempre foram construídas como uma coleção de dispositivos separados, independentes, e geridos individualmente. Mas a centralização é poderosa. É um princípio fundamental para a SDN, e é muito apropriado que se aplique a centralização para o software de rede. No entanto, não se pode levar isto muito longe, a centralização só faz sentido dentro de uma área geográfica confinada, altamente conectada, por exemplo, dentro de uma universidade, ao longo de uma cidade pequena ou, no caso de um prestador de

serviços, em uma cidade inteira. Mesmo com essa centralização, os dispositivos de rede em si, continuarão distribuídos e devem ter inteligência local.

Quando se adiciona o conceito de centralização para softwares de rede, os planos se deslocam um pouco. Independentemente do número de dispositivos distribuídos, o que se quer é gerenciar a rede como um sistema, e a Gestão Centralizada executa esse trabalho. Quando se centraliza a gestão, ela se torna a configuração mestre, todos os dispositivos mantêm apenas uma cópia. Isto é muito semelhante à maneira como publicações trabalham com *Smartphones* e *tablets*. Se for executado o aplicativo do Globo.com em um Smartphone, ele baixa a edição de hoje. Durante o dia, ele mantém a verificação de atualizações e baixa quando elas aparecem. Isso é análogo como funciona a Gestão Centralizada, o original fica no centro, e apenas uma cópia dos dados de configuração é armazenada em dispositivos de rede.

3.1 OPERABILIDADE DOS SERVIÇOS.

Os serviços, historicamente, têm sido implantados dentro de cada dispositivo de rede e de segurança, mas com a SDN, os Serviços podem se mover para o centro, sendo realizados em nome de todos os dispositivos. No entanto, isto só faz sentido em uma área geográfica contida e altamente conectada.

Quando a SDN entra, a imagem e algumas funções são tratadas de formas centralizadas, as alterações ao plano de Controle são as mais complexas. O plano de Controle é o policial que direciona o tráfego, a maneira que o plano de Controle trabalha é: cada dispositivo de rede fala com os dispositivos de rede com que ele se conecta diretamente. Cada dispositivo passa informações sobre a rede para o próximo dispositivo. Isso funciona muito bem em redes mundiais, altamente

conectada. Muitos anos de trabalho em toda a indústria de redes, garantem que as redes continuam executando o seu trabalho, mesmo quando as coisas não estão como deviam. Quando um roteador principal fica fora, há uma avalanche de pacotes de sinalização entre os dispositivos de rede, enquanto os mesmos tentam reestruturar sua visão da rede e manter os usuários ligados.

Mas às vezes ter uma vista panorâmica e central do tráfego também faz sentido. É onde o Controlador Centralizado entra em ação. O Controlador Centralizado tem uma visão ampla da rede e pode se conectar aos demais de uma maneira que aperfeiçoe o tráfego geral.

O Encaminhamento é um plano que sempre fica descentralizado em um mundo SDN. Isso faz sentido porque o Encaminhamento, na verdade, move os dados e esta ação é descentralizada, por definição.

3.1.1 **Migração dos Serviços**

Não podemos simplesmente ignorar as Redes Convencionais e implantar SDN como se não existissem antes, porque as redes estão ativamente em execução e devem continuar funcionando enquanto a SDN estiver sendo introduzida. A SDN é como uma remodelação: é necessário que se dê um passo de cada vez. Como a maioria das remodelações, há mais de uma maneira de se alcançar os resultados em SDN, mas aqui existe um razoável conjunto de passos que devem ser dados para que as metas sejam atingidas:

- O Gerenciamento é o melhor lugar para começar, pois, fornece a maior eficiência. O objetivo é centralizar a gestão da rede, análise, e configuração de funcionalidade para fornecer um único mestre, que configura todos os dispositivos de rede. Isto reduz custos operacionais e

permite aos clientes adquirir uma visão de negócios a partir de suas redes.

Muitos pontos envolvem o Gerenciamento Centralizado e cada um deles fornece um valor significativo. Semelhante aos aplicativos na nuvem, estes sistemas de Gestão Centralizada são empacotados em máquinas virtuais x86 (*VM*) e executados nos servidores padrão do setor. Esses *VM*s são conduzidos usando um dos sistemas de virtualização comumente disponíveis, como *VMware's vCloud Director*, *Microsoft System Center* ou *OpenStack*.

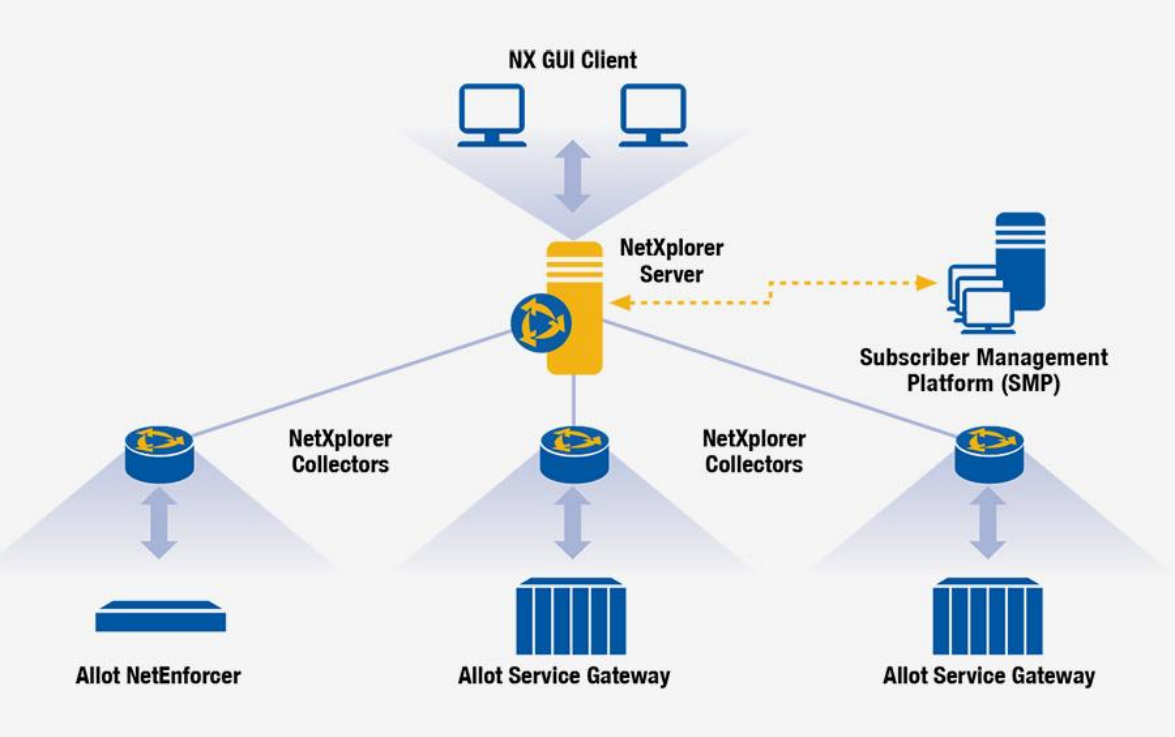

Figura 04 – Gerenciamento Centralizado. [25]

- Extrair os serviços dos dispositivos de rede e de segurança, criando *VMs* de serviços é um grande próximo passo, porque os serviços são áreas críticas e as vezes má distribuída pela rede. Permitindo que os serviços de rede e de segurança se multipliquem de forma independente usando

hardwares (padrão em Redes Convencionais), com base nas necessidades da solução.

Criar uma plataforma permitindo que os serviços a serem criados usem modernos VMs x86 abre um novo mundo de possibilidades. Por exemplo, a capacidade de um firewall de segurança atual é completamente limitada pelo alto desempenho de processamento aplicado na Rede, aplicar em um único dispositivo de rede o plano de Encaminhamento é significativamente mais eficiente, se for possível aplicar os serviços de segurança do dispositivo e em seguida, executá-los em uma estrutura de menor custo de servidores x86, aumenta drasticamente a capacidade e agilidade.

Logo, é possível criar um ponto de acesso, ou conectar esses serviços de volta a um dispositivo de rede único. Os servidores x86 podem ser colocados em um rack ao lado do dispositivo de rede ou podem ser implantados como lâminas de servidores (*Blade server*) dentro do mesmo dispositivo de rede. De qualquer forma, este passo abre as possibilidades de um novo conjunto de aplicativos de rede.

O Controlador Centralizado permite que múltiplos serviços de rede e segurança se conectem em série através de dispositivos dentro da rede. Isso é chamado de "Encadeamento de Serviço SDN". Utilizar o software para inserir serviços virtualmente no fluxo do tráfego de rede. A funcionalidade do serviço de encadeamento hoje é realizada fisicamente, usando dispositivos de rede e segurança separados. A abordagem física hoje do serviço de encadeamento é muito difícil, dispositivos separados estão ligados fisicamente por cabos Ethernet, cada dispositivo deve ser configurado individualmente para estabelecer a cadeia de serviços. Com o Encadeamento de Serviço SDN, redes podem ser reconfiguradas em tempo real, permitindo-lhes responder dinamicamente às necessidades do

negócio. O Encadeamento de Serviço SDN reduzirá drasticamente o tempo, o custo, e o risco para os clientes projetar, testar e fornecer novos serviços de rede e segurança.

Como os serviços estão desagregados dos dispositivos e as Cadeias de Serviço SDN estão estabelecidas, o hardware de rede e segurança pode ser usado para otimizar o desempenho, com base nas necessidades da solução. Hardware de rede e segurança continuarão a entregar 10x melhor o desempenho, comparado ao que pode ser realizado no software sozinho.

A separação dos planos ajuda a identificar a funcionalidade que é candidata à otimização dentro do hardware de Encaminhamento. Isto abre um potencial significativo de inovação dentro do ASIC e do projeto do sistema de dispositivos de rede e segurança. Enquanto um x86 é de uso geral, os ASICs dentro de dispositivos de rede são otimizados para encaminhar tráfego de rede em velocidades extremas.

Este hardware irá evoluir para tornar-se mais capaz a cada vez que mover se algo de software em um ASIC, será possível conseguir uma melhoria de desempenho de 10 x ou mais. Isto requer uma coordenação estreita entre o projeto ASIC, sistemas de hardware, e o software em si. Conforme a SDN tornar-se generalizada, a capacidade de otimizar o hardware irá criar muitas oportunidades para fornecedores de sistemas de segurança e de rede.

4 **OPENFLOW**

Baseado em informações dos capítulos anteriores, as Redes Definidas por Software estão cada vez mais em evidência, devido as estruturas atuais de Redes de Computadores não apresentarem boa escalabilidade no que tange as expectativas futuras da Internet. Separando o plano de controle e o plano de encaminhamento, se faz necessário uma interface para a troca de informação entre ambos os planos, onde a Controladora do Plano de Controle orienta os Dispositivos do plano de Dados ou Encaminhamento, como os pacotes devem ser tratos até os seus respectivos destinos, com isso o SDN estabelece novas formas para se desenvolver em ambientes virtuais.

O *OpenFlow* é uma das opções de interfaces lógicas que interligam o Plano de Controle e o Plano de Dados [5]. O protocolo *OpenFlow* foi desenvolvido para resolver os problemas em protocolos de redes legadas. Em arquitetura de rede definida por software (SDN) *com OpenFlow,* permite acesso direto a manipulação de plano de encaminhamento de dispositivos de rede, como switches e roteadores, tanto física como virtual.

Com o OpenFlow/SDN, os administradores podem personalizar as redes de acordo com as necessidades locais, eliminar ferramentas desnecessárias e criar redes virtuais e isoladas. Eles também podem aumentar o ritmo da inovação através do software, em vez do hardware, o que irá acelerar a troca de tecnologia com parceiros e a transferência de tecnologia entre universidades [6].

A estrutura de Redes Definidas por Software, funcionam em formato de comunicação Mestre/Escravo, onde os Switches OpenFlow são elementos desprovidos de inteligência, baseados em orientações estipuladas pelo Controlador OpenFlow. O Controlador é o responsável por manter a estrutura logicamente

escalável, após a estabilização de regras de fluxo e protocolos, os Switches OpenFlow aprendem as informações e tratam cada fluxo conforme o mestre (Controlador) mandou, mesmo que o Controlador esteja ausente, seja por falha ou manutenção, a Estrutura lógica da Rede permanecerá estável, porém, não havendo a possibilidade de escalar e nem alterar qualquer regra na tabela de fluxo.

Este cenário faz lembrar o início das Redes de Computadores, onde, os Mainframes eram o Cérebro de toda estrutura, os "Terminais Burros" (*Disk Less*), extraiam orientações centralizadas, a grande diferença está na inteligência do *Switch* OpenFlow em relação aos Terminas Burros que sem os concentradores (*Mainframes*), serviam apenas como grandes calculadoras cientificas, mesmo sendo considerados dispositivos de baixo ou quase nenhum nível de inteligência se comparando aos Controladores OpenFlow, são bem mais eficientes que seus antecessores totalmente "Burros".

O *OpenFlow* é implementado em ambos os lados da interface, entre os dispositivos da infraestrutura de rede e o software de controle SDN. Para identificar o tráfego de rede, o *OpenFlow* usa o conceito de fluxos com base em regras predefinidas que podem ser programadas estaticamente ou dinamicamente pelo software de controle, permitindo à SDN responder às mudanças em tempo real nos níveis de aplicação, usuário e de sessão. Nas redes legadas em uso, roteamento baseado em Protocolo de Internet (IP) não fornece esse nível de controle.

4.1 COMPONENTES DO SWITCH OPENFLOW

Um switch OpenFlow é composto por pelo menos três partes:

A tabela de fluxo, com uma ação associada com cada entrada de fluxo, para dizer o switch como processar o fluxo, um Canal Seguro que liga o switch à um processor

de controle remoto (O chamado Controlador), permitindo que os comandos e pacotes sejam enviados entre um controlador e comutador usando o protocolo OpenFlow. Um controlador *OpenFlow* conectado à um Switch *OpenFlow* via TCP ou TLS [7].

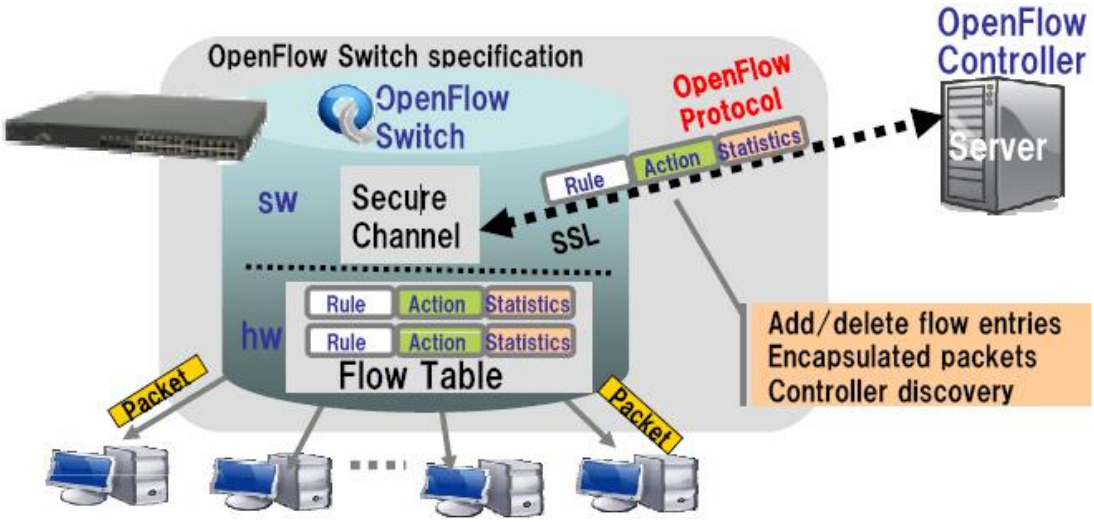

Figura 05 - Ambiente SDN/OpenFlow

4.1.1 Tabela de Fluxo

A entrada da tabela de fluxo é identificada por seus campos de correspondência e de prioridade: os campos de correspondência e de prioridade em conjunto identificam uma entrada de fluxo exclusiva na tabela de fluxo.

Segue a Tabela de Fluxo e seus respectivos campos:

Tabela 01 - Principais componentes de uma entrada na Tabela de Fluxo.

Match Fields	Counters	Actions

1. *Match Fields*: Conjunto de campos utilizados para equiparar (match) com os pacotes que chegam no switch. Podem ser campos dos cabeçalhos dos datagramas, porta de entrada ou até metadados previamente especificados.
2. *Counters*: Existem contadores específicos para tabelas, fluxos, portas ou filas que fornecem informações estatísticas. Por exemplo, o tempo (duração) que uma regra de fluxo foi instalada no switch, quantidade de pacotes ou bytes recebidos em uma determinada porta, ou por um determinado fluxo.
3. *Actions*: Conjunto de instruções que decretam como o switch deve manipular os pacotes recebidos que combinaram (match) com a regra de fluxo especificada. Caso nenhuma ação seja especificada, o pacote é descartado.

Cada entrada na tabela de fluxos do hardware de rede é composta por regras, ações e controles de estatística. A regra é formada pela definição dos valores de um ou mais campos do cabeçalho do pacote, é por meio dela que é determinado o fluxo. As ações então ficam associadas ao fluxo e vão determinar como os pacotes devem ser processados, para onde vão ser encaminhados ou se serão descartados. Os controles de estatística consistem de contadores utilizados para manter estatísticas de utilização e para remover fluxos inativos ou que não existam mais. Logo, as entradas da tabela de fluxos são interpretadas pelo hardware como decisões em cache do plano de controle em software, sendo, portanto, a mínima unidade de informação no plano de dados da rede [8].

Tabela 02 - Quadro usado na comparação n*a* Actions da Tabela de Fluxo.

In Port	*Ethernet*			VLAN		IP				TCP/UDP	
	src	dst	type	id	pcp	src	dst	proto	tos	srcp	dstp

- **In Port**: Porta de entrada do switch;
- **Ethernet Source**: Endereço MAC de origem;
- **Ethernet Destination**: Endereço MAC de destino;
- **Ethernet Type**: Tipo do quadro (frame) Ethernet;
- **VLAN ID**: Número de identificação da VLAN (Virtual LAN);
- **VLAN PCP (Priority Code Point)**: Nível de prioridade;
- **IP Source**: Endereço IP de origem;
- **IP Destination**: Endereço IP de destino;
- **IP Protocol**: Protocolo IP;
- **IP ToS (Type of Service)**: Tipo de serviço;
- **TCP/UDP Source Port**: Porta de origem do protocolo (TCP/UDP);
- **TCP/UDP Destination Port**: Porta de destino do protocolo (TCP/UDP) [9].

4.1.2 Canal Seguro

O protocolo OpenFlow descreve trocas de mensagens que ocorrem entre um controlador OpenFlow e um dispositivo OpenFlow. Normalmente, o protocolo é implementado em cima de *Secure Socket Layer* (*SSL*) ou *Transport Layer Security* (*TLS*), proporcionando um canal seguro ao OpenFlow.

O protocolo OpenFlow permite o controlador adicionar, atualizar e excluir ações para as entradas na tabela de fluxo [10].

Ele suporta três tipos de mensagens:

- **Controlador-de-dispositivo**: Estas mensagens são iniciadas pelo controlador e, em alguns casos, requerem uma resposta a partir do dispositivo.

- ***Asynchronous***: Estes tipos de mensagens são enviadas sem a solicitação do controlador.
- **Simétrica**: Estas mensagens são enviadas sem a solicitação a partir de qualquer controlador ou o dispositivo. Eles são simples, porém, uteis.

Tabela de mensagens entre o Controlador e os Dispositivos OpenFlow:

Tabela 03 - Tabela de Mensagens do OpenFlow [11].

Messages	**Description**
Controller-to-Device	
Features	Request the capabilities of a Switch. Switch Responds wuth a features reply that specifies its capabilities
Configuration	Set and query configuration parameters. Switch responds with parameter settings
Modify-state	Add, delete and modify flow/group entries and set switch port proprieties
Read-State	Collect information from switch, such as current configuration, statistics and capabilities
Packet-out	Direst packet to a specified port on the device
Barrier	Barrier request/reply messages are used by the controller to ensure message dependencies have been met or to receive notification for completed operations.
Role-Request	Set or query role of the OpenFlow channel. Useful when switch connects to multiple controllers.
Asynchronous - Configuration	Set filter on asynchronous messages or query that filter. Useful when switch connects to multiple controllers
Asynchronous	
Packet-in	Transfer packet to controller
Flow-removed	Inform the controller about the removal of a flow entry from a flow table.
Port-Status	Inform the controller of a change on a port.
Error	Notify controller of error or problem condition.
Symetric	
Helo	Exchanged between the switch and controller upon connection startup
Echo	Echo Request/reply messages can be sent from either the switch or the controller and they must return echo reply
Experimenter	For additional functions.

Este elemento não só garante que uma rede SDN não sofra ataques de elementos mal-intencionados como garante também que a troca de informações entre os comutadores e controladores da rede sejam confiáveis com baixa taxa de erros. A interface de acesso que o projeto do OpenFlow recomenda é o SSL (Secure Socket Layer), no entanto existem outras alternativas como o TCP e o PCAP (packet capture) sendo muito úteis em ambiente virtuais e de experimentação por sua simplicidade de utilização.

4.2 CONTROLADOR

Tem a finalidade de manipular a tabela de fluxo dos dispositivos *OpenFlow* decidindo o melhor caminho de cada aplicação, o controlador se comunica com o equipamento usando o protocolo *OpenFlow* através de um canal seguro. O *OpenFlow* não é o único protocolo para este tipo de comunicação, porém, o único implantado em ambientes de Produção.

A manipulação da Tabela de Fluxo, pode ser feita estaticamente ou dinamicamente, a forma como deve ser configurada está proporcionalmente ligada ao Hardware denominado a esta função.

Por exemplo, um Controlador estático pode ser uma aplicação simples rodando em um PC para estaticamente estabelecer fluxos que interliguem um conjunto de computadores de teste durante um experimento. Outro caso que podemos imaginar seriam controladores mais sofisticados que podem dinamicamente adicionar / remover fluxos enquanto a experiência progride[12].

Para que haja conexão entre dois dispositivos de Redes, seja de camada dois ou de camada três, há necessidade da troca de informações de controles e sinalização, com OpenFlow não é diferente, segue abaixo uma tabela com as

devidas mensagens, informando o sentido e descrição das mesmas, para que o Controlador e a Tabela de Fluxo dos dispositivos possam estabelecer conexão e trocar dados:

Tabela 04 - Mensagens trocadas pelo Controlador e Switch [13].

Mensagem	Tipo	Descrição
Hello	Controller->Switch	em seguida ao handshake TCP, o controlador envia o seu número de versão para o switch.
Hello	Switch->Controller	o switch responde dizendo o seu número de versão.
Features Request	Controller->Switch	o controlador pede para ver quais portas estão disponíveis.
Set Config	Controller->Switch	neste caso, o controlador pede ao switch para enviar fluxos que expirarem.
Features Reply	Switch->Controller	o switch responde com uma lista de portas, velocidades de portas, e tabelas e ações suportadas.
Port Status	Switch->Controller	permite ao switch informar ao controlador sobre mudanças em velocidades das portas e conectividade. Ignore essa mensagem por agora, e um recurso instável.

Os controladores fornecem *APIs* (*Application Programming Interface*) para que sejam desenvolvidos protótipos de Redes Definidas por *Software*. Para o protocolo OpenFlow, os principais controladores são o NOX e o POX, desenvolvidos pelos criadores do protocolo OpenFlow. A diferença entre eles é que o NOX utiliza a linguagem C++ e o POX, python. Há outros controladores para o protocolo OpenFlow, como, por exemplo, Trema[26], Maestro[27], Beacon[28], FML [29] e Frenetic [30] [14].

Segue abaixo uma relação de Controladoras e suas especificações em programação:

Tabela 05 – Controladoras e suas especificações de programação [15].

Controladores e suas especifições	
POX	(Python) Pox as a general SDN controller that supports OpenFlow. It has a high-level SDN API including a queriable topology graph and support for virtualization.
IRIS	(Java) a Resursive SDN Openflow Controller created by IRIS Research Team of ETRI. Our vision was to create an SDN controller platform with the following features : (a) Horizontal Scalability for carrier-grade network (b) High Availability with transparent failover from failure (c) Multi-domain support with recursive network abstraction based on Openflow
MUL	(C) MūL, is an openflow (SDN) controller. It has a C based muli-threaded infrastructure at its core. It supports a multi-level north bound interface for hooking up applications. It is designed for performance and reliability which is the need of the hour for deployment in mission-critical networks.
NOX	(C++/Python) NOX was the first OpenFlow controller
Jaxon	(Java) Jaxon is a NOX-dependent Java-based OpenFlow Controller.
Trema	(C/Ruby). Trema is a full-stack framework for developing OpenFlow controllers in Ruby and C
Beacon	(Java) Beacon is a Java-based controller that supports both event-based and threaded operation
Floodlight	(Java) The Floodlight controller is Java-based OpenFlow Controller. It was forked from the Beacon controller, originally developed by David Erickson at Stanford.
Maestro	(Java) Maestro is an OpenFlow "operating system" for orchestrating network control applications
NDDI - OESS	OESS is an application to configure and control OpenFlow Enabled switches through a very simple and user friendly User Interface.
Ryu	(Python) Ryu is an open-sourced Network Operating System (NOS) that supports OpenFlow
NodeFlow	(JavaScript) NodeFlow is an OpenFlow controller written in pure JavaScript for Node.JS
ovs-controller	(C) Trivial reference controller packaged with Open vSwitch.

4.3 INTERFACES OPENFLOW NORTHBOUND API E SOUTHBOND API.

Em Redes Definidas por Softwares o OpenFlow tem se destacado bastante em relação aos seus concorrentes, devidamente pelos resultados que vem apresentando em pouco tempo de evolução, além das Universidades e pesquisadores Autônomos, existem organizações que foram criadas exclusivamente para tratar desse assunto, de forma Operacional e comercial, trazendo cada vez mais à tona o OpenFlow como se fosse não houve competitividade comercial.

A ONF (Open Networking Foundation) é uma organização orientada para o utilizador dedicado à promoção e adoção de Rede Definidas por Software (SDN), através do desenvolvimento de padrões abertos.

ONF enfatiza um processo de desenvolvimento aberto e colaborativo que é conduzido a partir da perspectiva do usuário final. Nossa maior conquista até agora é introduzir o Padrão OpenFlow®, que permite a programação remota do plano de encaminhamento [16].

Além de divulgações feitas por Órgãos e Instituições de pesquisa, o *OpenFlow* cumpri com o que se propõe em relação a tratamento de Fluxo, melhor escalabilidade e aplicação de serviços, vimos que a separação do plano de Dados ou Encaminhamento e o plano de Controle dos dispositivos *OpenFlow* fizeram toda diferença no fluxo de Dados, permite melhor eficiência de ambos os planos, principalmente no Plano de Encaminhamento, que se encarrega de analisar as regras vindo do Controlador, comparar os cabeçalhos e entregar dados ao destinatário ou dispositivo mais próximo. O *OpenFlow* usa *APIs* que auxiliam ainda mais na organização do fluxo, tratando-os em cada sentido de forma dedicada, fazendo uso das Interfaces *Northbound* e Interfaces *Southbound*.

4.3.1 **Interfaces Northbound**

Em uma arquitetura de rede definida por software (SDN), as interfaces de programação de aplicações (APIs) *Northbound* são usados para a comunicação entre o controlador de SDN e os serviços e aplicativos em execução na rede. As APIs *Northbound* podem ser utilizadas para facilitar a inovação e permitir orquestração e automação da rede eficaz para o alinhamento com as necessidades das diferentes aplicações em SDN [17].

4.3.2 **Interfaces Southbound**

Sua principal função é permitir a comunicação entre o controlador de SDN e os nós de rede (ambos os switches e roteadores virtuais e físicos) para que o roteador pode descobrir a topologia da rede, definem os fluxos de rede e implementar as solicitações relacionadas a ele por meio de APIs Northbound [18].

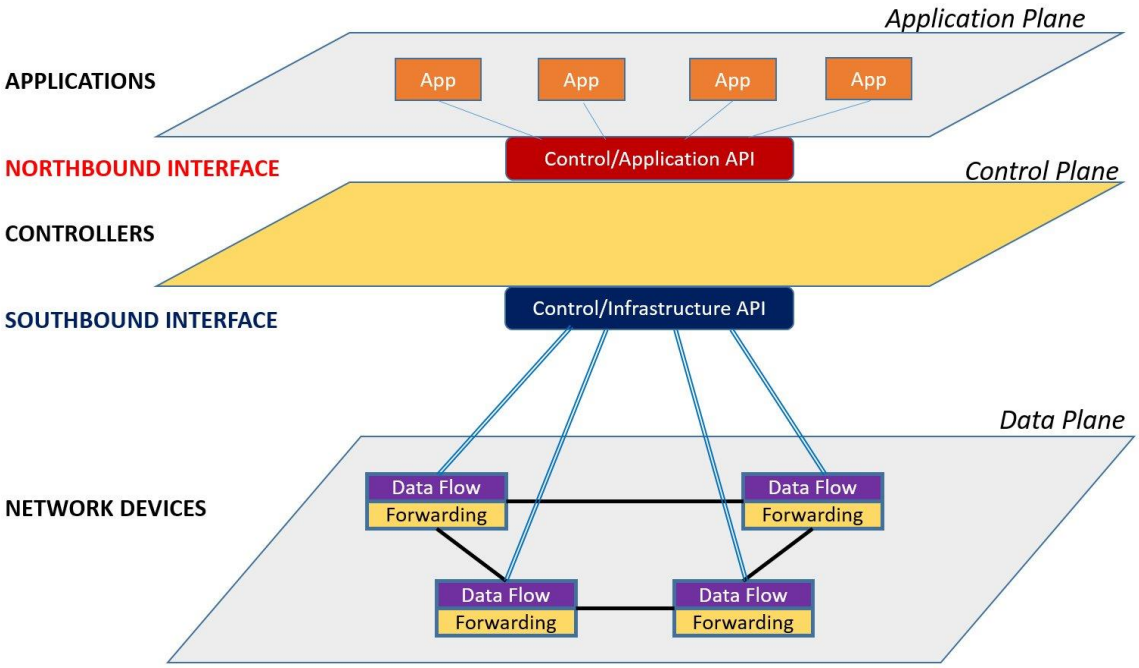

Figure 1 - Software-Defined Networking – A high level architecture

Figura 06 - Interfaces API Northbound e Southbound [19].

Com uma API geral como o OpenFlow, a SDN torna muito mais simples a introdução de novos fornecedores de sistemas operacionais. Permite que os utilizadores criem plug-ins para adicionarem características ao plano de controle sem terem de modificar o hardware fundamental ou melhorar o hardware sem modificar o plano de controle [6].

5 INCOMPATIBILIDADES

Desde o início as Redes de Computadores apresentam algumas incompatibilidades entre produtos de Fabricantes diversos.

A fim de resolver as incompatibilidades entre fabricantes, na década de 1970 a ISO (*International Organization* for *Standardization*) criou um padrão universal para troca de informações entre e dentro das redes e também por meio de fronteiras geográficas. Esse padrão para arquitetura de redes era o Modelo de Referência OSI, estabelecido em sete camadas, o qual incentivou a padronização de redes de comunicação e controle de processos distribuídos [20].

Ainda com toda padronização, hoje os fabricantes criam seus próprios métodos (em paralelo com os protocolos abertos e homologados), a fim de tentar monopolizar o mercado de dispositivos de Redes, tentando aficadamente criar suas caixas magicas que resolvam todos os problemas de uma estrutura de Redes.

Existem empresam que apresentam equipamentos, qual são vendidos como Switch de Camada 7 (L7), disponibilizando funções de firewall e capacidade para atender como um servidor de backup, garantindo 99,999% de SLA de todos os serviços, mesmo quando todos atuam em conjunto. Essa mesma empresa ganhou espaço no mercado devido a um único produto que atende com Balanceador de Cargas e se tornaram atraentes devido ao custo e benefício, e sabemos que Switches são dispositivos de camada 2 (L2).

Antes mesmo do sucesso do monopólio, os fabricantes na verdade acabam dificultando o conhecimento técnico, com as suas particularidades e quando operam em conjunto tendem aumentar ainda mais o tempo de propagação, atrasos de Buffers e processamento, devido a essas incompatibilidades.

5.1 CASOS DE INCOMPATIBILIDADES.

Todo este trabalho tem como objetivo tratar a falta de escalabilidade das Redes Convencionais com os novos recursos que as Redes Definidas por Software nos apresentam, sabemos que a distância entre o nó Origem e Destino, impactam de forma diretamente proporcional no encaminhamento do pacote, devido à enorme possibilidade de o percurso possuir as mais diferenciadas *medias* possíveis (Como Cabo par trançado, Fibra Ótica, ondas de rádio e etc.), conectadas a outros diversos dispositivos com aplicações e funcionalidades diversificadas e seus respectivos protocolos de controle, sinalização e propagação de sinal.

A seguir veremos um cenário onde há diversos fabricantes e será comparado algumas soluções diferenciadas a fim de tratar o mesmo problema.

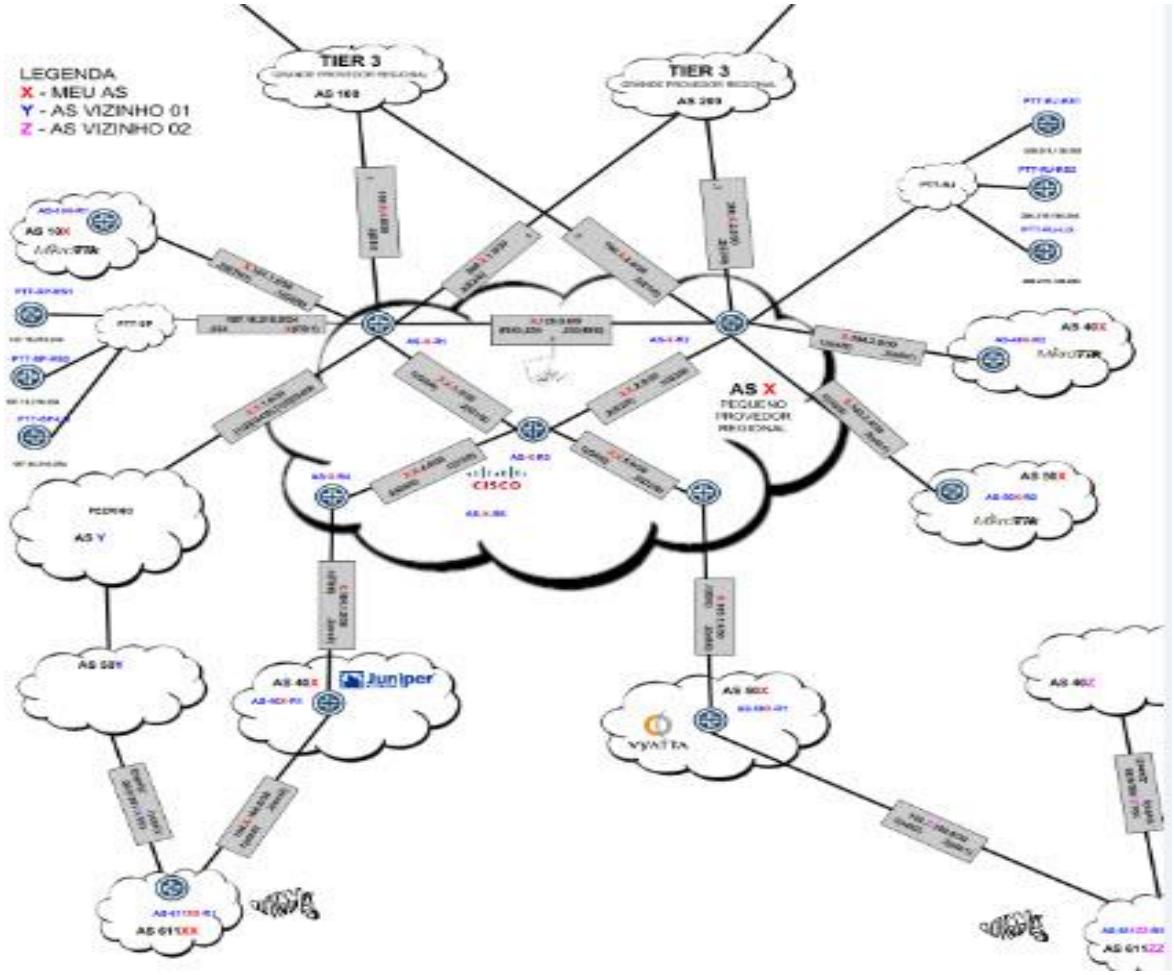

Figura 07 – Ambiente Multi-Vendor

5.1.1 **Tabelas de Roteamento entre Fabricantes.**

Há uma série de diferenças no comportamento padrão para anunciar rotas inativas por BGP.

As rotas inativas são rotas que não estão instalados no RIB (não selecionado como o melhor caminho), na maioria das vezes, porque eles também são aprendidos a partir de outro protocolo de roteamento que tem uma melhor distância administrativa (Cisco) ou preferência de rota (terminologia *Juniper)*.

Como uma revisão desses valores, segue tabela comparando as métricas como: AD da Cisco e Preferência de Rota Juniper para alguns dos protocolos de roteamento:

Tabela 05 – Comparação das Distancias Administrativas entre Cisco e Juniper.

Route Source	Cisco Administrative Distance	Juniper Router Perference
Connected	0	0
Static	1	5
EIGRP	Summary 5	N/A
	Internal 90	
	External 170	
OSPF	110	Internal 10
		External 150
IS-IS	115	Level 1 internal 15
		Level 2 internal 18
		Level 1 external 160
		Level 2 external 165
RIP	120	100
BGP	Internal (iBGP) 200	170
	External (eBGP) 20	

5.1.2 **Caso de *Vlans* Incompatíveis.**

A Cisco Implementa uma propriedade no Protocolo *Spannig Tree*, que é Per-VLAN na essência. Isto dá origem a certas considerações em um ambiente *multi-vendor*. Quaisquer terceiros Switches que implementam STP baseado no padrão IEEE, tem uma implementação de *Spanning* Tree que tem uma única instância por Switch, independentemente do número de *VLANs* no local. Conectar esses Switches em um ambiente Cisco pode resultar em topologias STP quebradas com múltiplos *root bridges* (Switches na topologia Spanning-Tree, que gerenciam as *BPDU's)*, o que não garante a interoperabilidade entre dispositivos. No entanto, alguns fornecedores (Extreme Networks, Force10, Brocade e Juniper RSTP) possuem switches/implementações que interagem com Endereço MAC e STP Cisco para garantir a interoperacionalidade [21].

Em Redes Definidas por Softwares todas essas incompatibilidades são tratadas de forma transparentes, tendo o Protocolo *OpenFlow* como o "único" padrão a ser considerado na Rede, todos os outros protocolos da pilha TCP/IP, passam a ser apenas parâmetros que devem ser comparadas na Tabela de Fluxo do Switch *OpenFlow*, com Exceção da camada de Transporte que continuará efetuando a sua função (comunicação Fim-a-Fim), inclusive entre o Controlador OpenFlow e o Switch OpenFlow, suportando o Canal Seguro para que os Elementos *OpenFlow* se comuniquem e compartilhem regras/funções.

Quaisquer fabricantes que desejam se manter e/ou participar ao mercado futuramente, deverão encarar como desafio a homogeneidade do *OpenFlow*, por enquanto, inicialmente o comércio dos hardwares continuarão bastante competitivos, atraindo fabricantes como HP e DELL, que são destaques nesse mercado e Empresas como a VMware, especializadas em virtualização.

6 CONCLUSÃO

O intuito deste trabalho visa comparar as Redes Definidas por Softwares e as Redes Legadas, tento como perspectiva mostrar os benefícios de migrar para as novas estruturas que apesar de virtualizadas, se destacam em escalabilidade, são mais flexíveis, robustas em relação à processamento (devido a independência do plano de Controle) e resiliente em relação a novos ambientes.

Apontando a incompatibilidade de equipamentos entre fabricantes, proporcionada pelas suas particularidades na forma de tratar Dados na Rede, cada fabricante desenvolvendo soluções cada vez mais isoladas, usando parâmetros próprios como base de informações para melhorar o desempenho da Rede ou facilitar o troubleshooting, na maioria, são recursos bastante eficientes, mas a necessidade de atuar em ambientes com único fabricante, tornam os recursos inviáveis.

Apesar de existir outros protocolos similares ao OpenFlow, este é quem melhor se adequou e vem sendo apresentado pelas próprias Redes Definidas por Software, como o mais estável e garantido de todos *OpenSource*.

O OpenFlow teve seu destaque no trabalho devido a enumeras soluções em Redes Definidas Por Software, o protocolo inovou a forma com os dispositivos, protocolos e pacotes são tratados.

Administrativamente falando, outros tópicos como: API, linguagem de programação C ++, Python e Linux, agora fazem parte do currículo do profissional de Redes, tanto quanto os protocolos de Roteamento e de enlace pertencem aos desenvolvedores de Redes. Resumindo, ESCALABILIDADE é a palavra-chave desse trabalho de modo geral.

Para novas estruturas de Redes, novos serviços e funcionalidades mais abrangentes, com a tendência de virtualização em massa, as Redes Definidas por Softwares constroem estruturas econômicas e eficientes, que atendem alto fluxo dados de forma mais inteligente, e as convergências, seja por falha ou implantação, mais ágeis, por conta da centralização do Plano de Controle. Em conjunto outro conceito chamado NFV (*Network Functions Virtualization*) [32] trata os serviços de Redes de modo virtual, substituindo maquinas físicas e especificas por servidores de alta capacidade suportando *VMs,* onde, expandir a estrutura ou fazer upgrades, não necessariamente significa aumentar custos com mais elementos ou aluguel de espaços em DataCenters.

Controlar os serviços dos clientes será uma tarefa mais branda, as Operadoras serão capazes criar pacotes de serviços mais completo, de acordo com o portfólio cliente. Uma fibra pode ser entregue com tudo que o usuário necessita, seja QoS para serviços especializados, controle de fluxo com demandas especificas e segurança (Proxy, Firewall e regras).

Estamos progredindo constantemente, evolução dos grandes Datacenters, informações centralizadas por aplicações e serviços melhor distribuídos por Operadoras, transporte de dados nas escalas de PETA, EXA, ZETTA e YOTTA Bytes, mobilidade e acesso quase ilimitada, contudo, para que possamos evoluir definitivamente, se faz necessário, usuários e profissionais que também procuram pela evolução interna, caso contrário, todo o progresso tecnológico se tornará ferramenta de destruições em massa, por crenças e valores individuais.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Tanenbaum, A. S. **Redes de Computadores: Quarta Edição**, Editora Campus, 2003.

[2] Jackson, j. **VMware to Acquire OpenFlow Pioneer Nicira for $1.26 Billion.** Disponível em: **<**http://www.pcworld.com/article/259705/vmware_to_acquire_openflow_pioneer_nicira_for_126_billion.html>. Acesso em 9 Dez. 2015.

[3] KLEIS, E. G. **Redes Definidas por SW I: Aplicação para Diferenciação de Caminhos**. Dísponivel em: <http://www.teleco.com.br/tutoriais/tutorialsw1/default.asp>. Acesso em: 4 Dez, 2015.

4] BRITO, S. H. B. **Paradigma SDN de Redes Programáveis,** Disponível em: <http://labcisco.blogspot.com.br/2013/07/paradigma-sdn-de-redes-programaveis.html>, Acesso em: 10 Jan, 2016.

[5] SDXCENTRAL. What is OpenFlow? Definition and how it relates to SDN. Disponível em : <https://www.sdxcentral.com/resources/sdn/what-is-openflow/>. Acessado em: 3 jan. 2016.

[6]. **IDGNS**. **OpenFlow e SDN: o futuro das redes.** Disponível em: <http://www.computerworld.com.pt/2013/12/02/openflow-e-sdn-o-futuro-das-redes/>. Acessado em: 3 jan. 2016.

[7] . McKeown, N.; ANDERSON, T.; BALAKRISHNAN, M.; PARULKAR, G.; PETERSON, L.; REXFORD, J.; SHENKER, S.; TURNER, J. **OpenFlow: Enabling Innovation in Campus Networks.** disponível em: <http://archive.openflow.org//documents/openflow-wp-latest.pdf>. Acessado em: 3 jan. 2016.

[8]. COSTA, L. R., **OpenFlow e o Paradigma de Redes Definidas por Software**. Disponível em: <http://monografias.cic.unb.br/dspace/bitstream/123456789/391/1/Monografia_Vesao_Leitura_em_PC.pdf>. Acessado em: 19 nov, 2015.

[9]. HELLER, B., **OpenFlow Switch Specification Version 1.0.0.** 2009. Disponível em: <http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf>. Acessado em: 13 out. 2015.

[10]. THE RANDOM SECURITY GUY**. Openflow**. Disponível em: <http://therandomsecurityguy.com/openflow/>. 2014. Acessado em: 19 nov. 2015.

[11]. AGARWAL, K. S., **Understanding OpenFlow.** 2014. Disponível em: <https://sdngeeks.wordpress.com/2014/09/01/understanding-openflow/>. Acesso em: 28 set. 2015.

[12]. MARCONDES, C., **Projeto de Desenvolvimento em OpenFlow**. 2011. Disponível em: <http://www.inf.ufes.br/~magnos/IF/if_files/Tutorial.pdf>. Acessado em: 3 jan. 2016.

[13]. FIBRE PROJECT. **OPENFLOW.** Disponível em: <http://vineg.wikispaces.com/file/view/doc-openflow.pdf>. Acessado em: 28 set. 2015.

14] LOBATO, A.; FIGUEIREDO, U.; ALVES, L. **Redes Definidas por Software.** 2013. Disponível em: <http://www.gta.ufrj.br/grad/13_1/sdn/principaisFerramentas.html>. Acessado em: 3 jan. 2016.

[15] CASADO, M., **List of OpenFlow Software Projects.** Disponível em: <http://yuba.stanford.edu/~casado/of-sw.html>. Acessado em: 3 jan. 2016.

[16] OPENNETWORKING.**ONF Overview.** Disponível em: <https://www.opennetworking.org/about/onf-overview>. Acessado em 9 dez. 2016.

[17] SDXCENTRAL**. What are SDN Northbound APIs?.** Disponível em**:** <https://www.sdxcentral.com/resources/sdn/north-bound-interfaces-api/>. Acessado em: 3 jan.2016.

[18]. GIBILISCO, S,. **Northbound interface / Southbound interface.** 2012. Disponível em: <http://whatis.techtarget.com/definition/northbound-interface-southbound-interface>. Acessado em: 3 jan. 2016.

[19]. HOANG, D,. **Software Defined Networking – Shaping up for the next disruptive step?.** 2015. Disponível em: <http://telsoc.org/ajtde/2015-12-v3-n4/a28>. Acessado em: 10 jan, 2016.

[20]. MENDES, D. R., **Redes de Computadores - 1ª Edição. 2007. Disponível em:** <http://novatec.com.br/livros/redescom/>. Acessado em: 13 out. 2015.

[21]. MANSUR, H., **Cisco and Force10 – STP Spanning Tree Interoperability.** 2012. Disponível em: <https://hasanmansur.com/2012/10/15/cisco-and-force10-stp-spanning-tree-interoperability/>. Acessado em: 13 out. 2015.

[22] CISCO. **Understanding Control Plane Protection. Disponível em**: http://www.cisco.com/c/en/us/about/security-center/understanding-cppr.html. Acessado em:3 jan.2016.

[23] CELLO, Marco. **Software Defined Networking (SDN)**. 2014. Disponível em:https://view.officeapps.live.com/op/view.aspx?src=http://www.ieiit.cnr.it/files/Presentazione_Cello_SDN.ppt. Acessado em: 3 jan.2016.

[24] DOYLE, L. **The return of the network operating system (NOS)**. 2013. Disponível em**:** http://www.networkworld.com/article/2162773/lan-wan/the-return-of-the-network-operating-system--nos-.html**.** Acessado em: 19 nov. 2015.

[25] SERVNET. **Gerenciamento Centralizado.** Disponível em:www.servnet.inf.br/vídeos-trustwave/28-protudos/allot. Acessado em:14 mar. 2016.

[26] DIETZ, T. **Trema Tutorial**. 2012. Disponível em: http://www.fp7-ofelia.eu/assets/Uploads/201203xx-TremaTutorial.pdf**.** Acesso em: 14 mar. 2016.

[27] CAI, Z.; COX, A. L.; NG, T. S. E. **Maestro: A System for Scalable OpenFlow Control.** Disponível em**:<** https://www.cs.rice.edu/~eugeneng/papers/TR10-11.pdf**>.** Acesso em: 14 mar. 2016.

[28] ERICKSON, D. **The Beacon OpenFlow Controller.** Disponível em: http://dl.acm.org/citation.cfm?id=2491189, acessado em: 14 mar. 2016.

[29] HINRICHS, T. L.; GUDE, N. S.; CASADO, M.; SHENKER, S.; MITCHELL, J. C. **Practical Declarative Network Management.** Disponível em: **http://conferences.sigcomm.org/sigcomm/2009/workshops/wren/papers/p1.pdf.** Acessado em: 14 mar. 2016.

[30] FRENETIC-LANG. **Python + Frenetic = Pyretic.** Disponível em: http://frenetic-lang.org/pyretic/. Acessado em: 14 mar. 2016.

[31] CISCO. **Internert of Things.** Disponível em: http://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html. Acessado em: 15 mar. 2016.

[32] MARVÃO, S. **SDN e NFV revolucionarão a operação das telecomunicações** Disponível em: http://www.bitmag.com.br/2014/09/sdn-e-nfv-revolucionarao-operacao-das-telecomunicacoes/#qDOQhgYCkZtG0faP.99. Acessado em: 15 mar. 2016.